idc15

Imagination Developers Connection

PowerVR Graphics Keynote

Rys Sommefeldt

PowerVR Rogue Hardware

PowerVR Rogue Recap

Optional secondary message

- **Formally announced at CES 2012**
- **Tile-based deferred renderer**
  - Building on technology proven over 5 previous generations
- **USC - Universal Shading Cluster**
  - New scalar SIMD shader core
  - General purpose compute is a first class citizen in the core …
  - … while not forgetting what makes a shader core great for graphics

TBDR

Tile-based

- **Tile-based**
  - Split each render up into small tiles (32x32 for the most part)
  - Bin geometry after vertex shading into those tiles
  - Tile-based rasterisation and pixel shading
  - Keep all data access for pixel shading on chip

TBDR

Deferred

- **Deferred rasterisation**
  - Don’t actually get the GPU to do any pixel shading straight away
  - HW support for fully deferred rasterisation and then pixel shading
  - Rasterisation is pixel accurate

End result

- **Bandwidth savings across all phases of rendering**
  - Only fetch the geometry needed for the tile
  - Only process the visible pixels in the tile

- **Efficient processing**
  - Maximise available computational resources
  - Do the best the hardware can with bandwidth

Power

- **Maximising core efficiency**
  - Lighting up the USC less often is always going to be a saving

- **Minimising bandwidth**
  - Texturing less is a fantastic way to save power
  - Geometry fetch and binning is often more than 10% of per-frame bandwidth
  - Saves bandwidth for other parts of your render

Rogue USC

Architectural Building Block

- **Basic building block of the Rogue architecture**
- **Laid out in pairs, with a shared TPU**
- **1, 0.5 and 0.25 USC designs are special**
  - Different balance in the design
  - Tend to find their way into non-gaming applications

Rogue USC

Shader Architecture

- **16-wide in hardware**
- **32-wide branch granularity**
    - We run half a task/warp per clock
- **Scalar SIMD**
- **Optimised ALU pipeline**
    - Mix of F32, F16, integer, floating point specials, logic ops

Rogue USC

Pipeline datapaths

- **Configurable in the IP core**
  - F16 paths were sometimes optional, thankfully not any more
  - F16 paths performance increased significantly after the first generation

- **Performance in your shader**
  - F32 paths are dual FMAD
  - F16 paths can do different things per cycle depending on shader
  - All up to the compiler
  - ISA is available for you to interrogate though, with disassembling compilers

Rogue USC - Scalar

- Hard to understate what a benefit this is
- Seems obvious to do, right?
- Vector architectures are just hard to program well
- Scalar isn’t a free lunch
- Makes performance a lot more predictable for you

Rogue USC

Programmable output registers

- **The pixel output registers in the ISA are read/write**
- **One per pixel**
- **Width depends on IP core**
- **We expose it programmatically with Pixel Local Storage**
  - Worked closely with ARM (thanks, Jan-Harald!)
    - Doesn’t matter that we hate their guts*, still need to make life great for developers

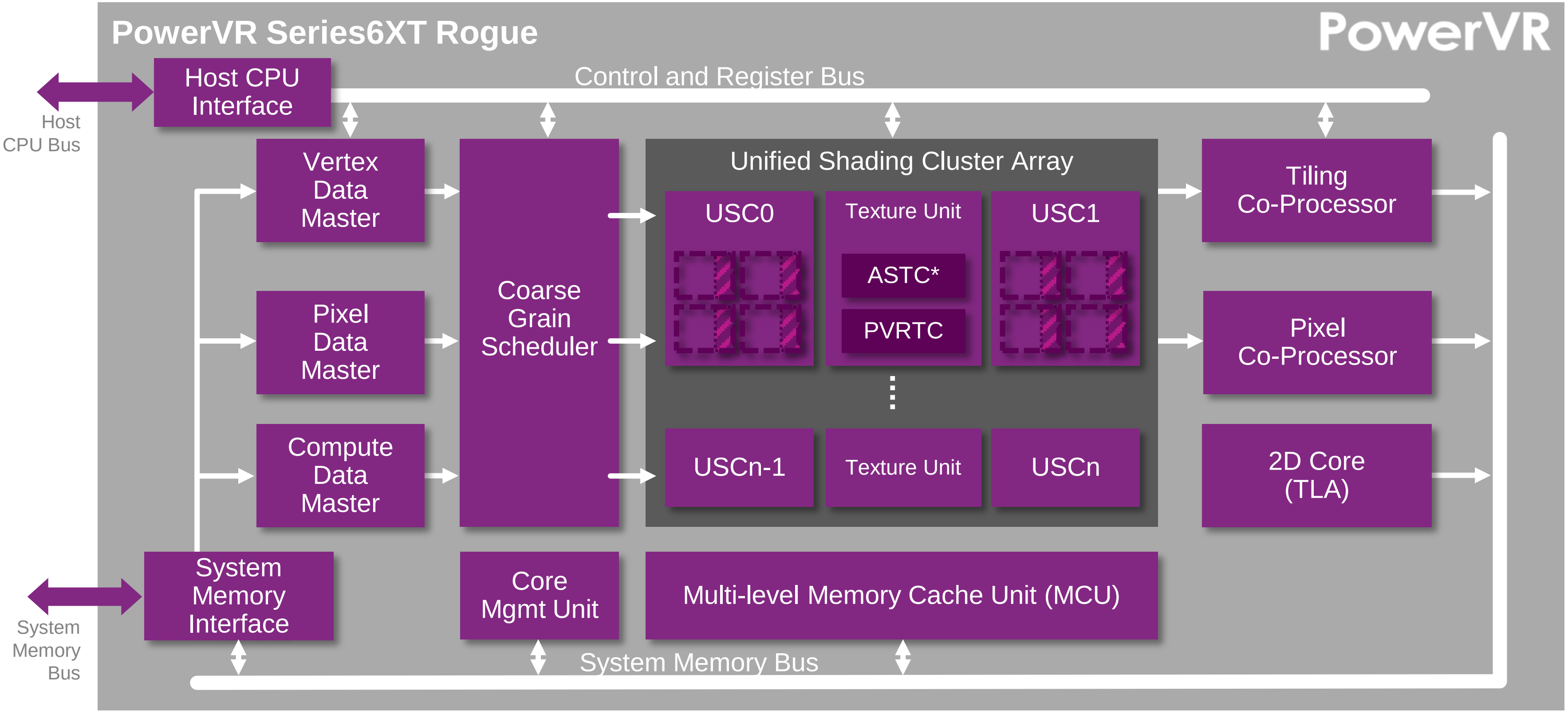
* We don’t really <3

Evolution

Health Warning: Really Bad Diagrams™

Rogue Evolution

- Architecture has changed quite a bit over time
- Rogue in 2010 still mostly looks like a Rogue today
- Significant evolutionary changes across the architecture
- Lots of it driven by developers before the IP is baked
- Lots of it driven by also analysing your stuff anyway



PowerVR Series6XT Rogue

PowerVR

Host CPU Bus

Host CPU Interface

Control and Register Bus

Vertex Data Master

Pixel Data Master

Compute Data Master

Coarse Grain Scheduler

Unified Shading Cluster Array

USC0

Texture Unit

ASTC*

PVRTC

USC1

USCn-1

Texture Unit

USCn

Tiling Co-Processor

Pixel Co-Processor

2D Core (TLA)

System Memory Bus

System Memory Interface

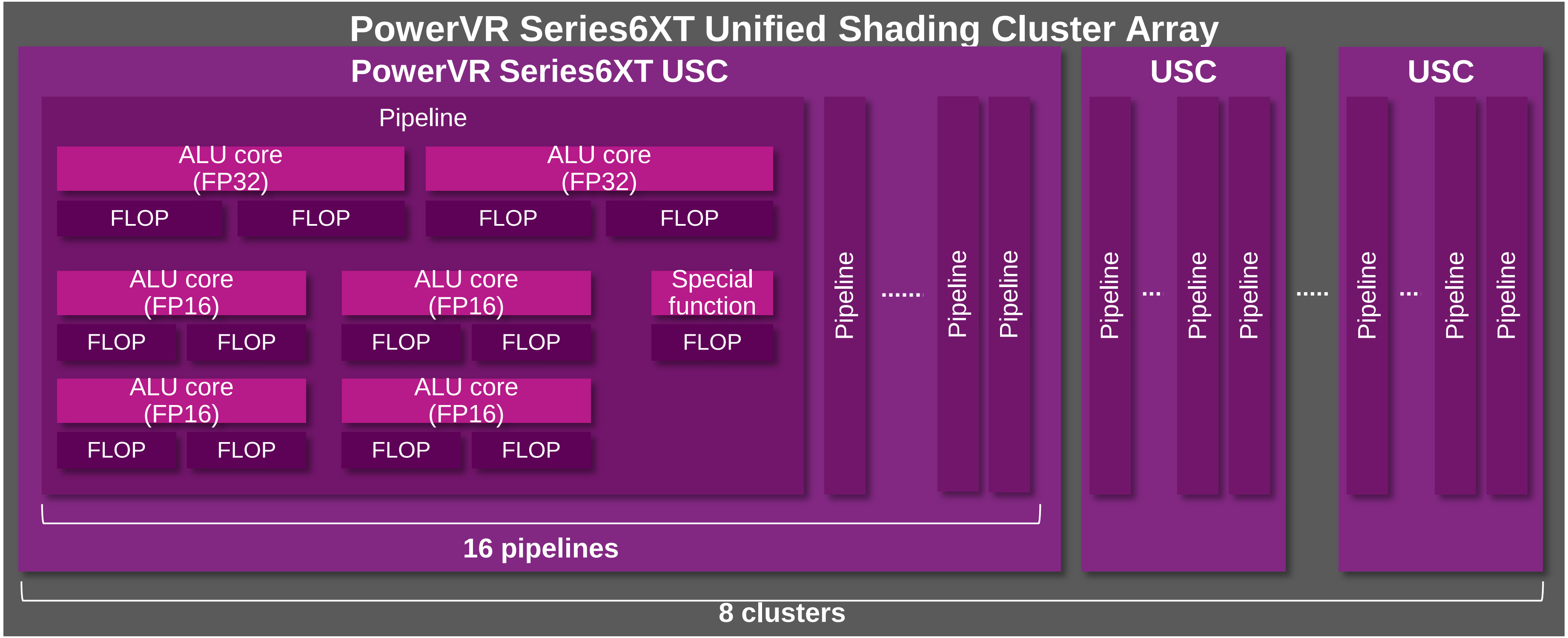
Core Mgmt Unit

Multi-level Memory Cache Unit (MCU)

System Memory Bus

▨ ***Extra low power GFLOPS***

* ***Supports both LDR and HDR ASTC formats***



PowerVR Series6XT Unified Shading Cluster Array

PowerVR Series6XT USC

Pipeline

ALU core (FP32)

FLOP FLOP

ALU core (FP32)

FLOP FLOP

ALU core (FP16)

FLOP FLOP

ALU core (FP16)

FLOP FLOP

Special function

FLOP

ALU core (FP16)

FLOP FLOP

ALU core (FP16)

FLOP FLOP

Pipeline ...... Pipeline Pipeline

16 pipelines

USC

Pipeline ... Pipeline Pipeline

.....

USC

Pipeline ... Pipeline Pipeline

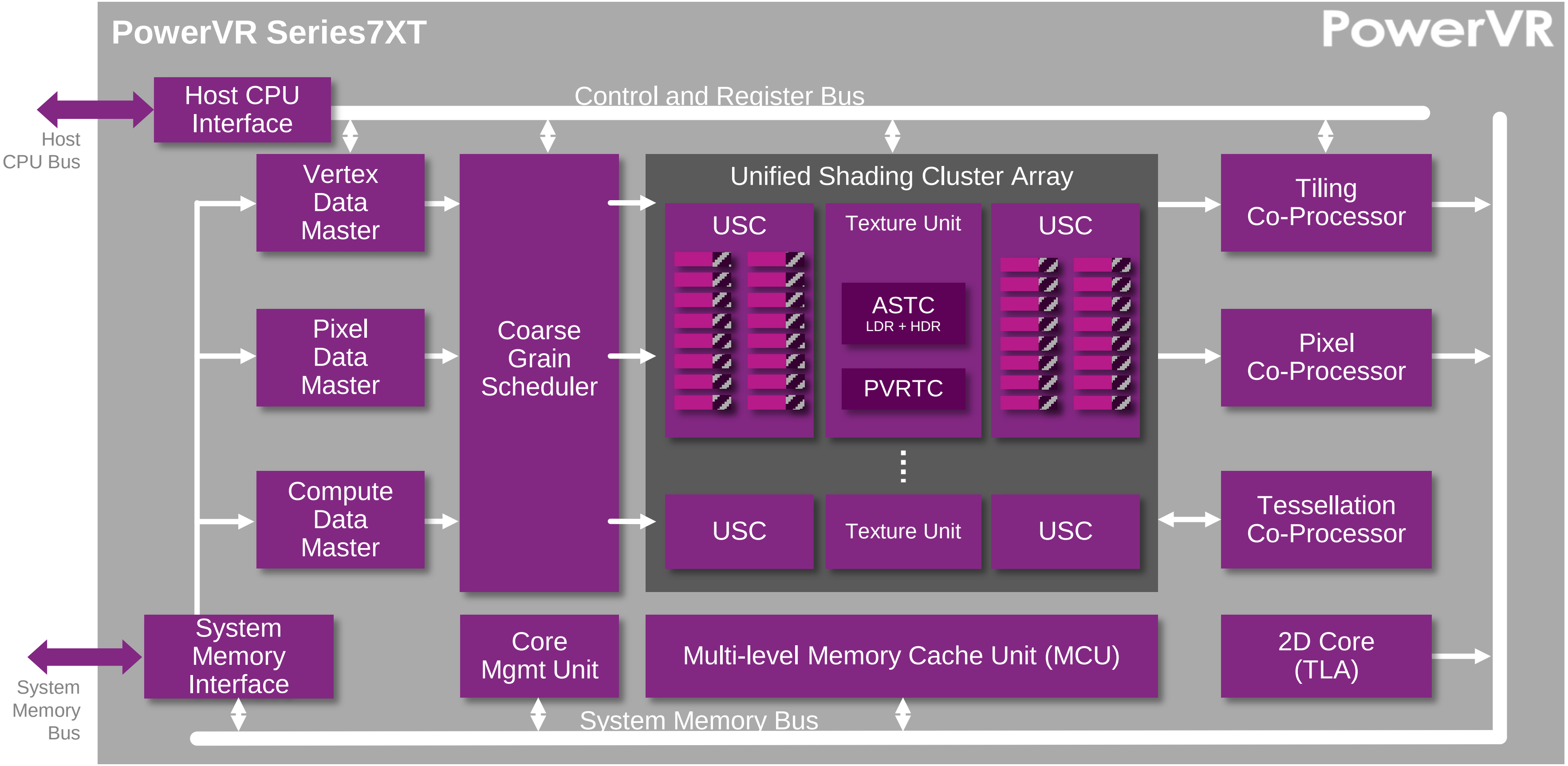
8 clusters

Series6 to Series6XT

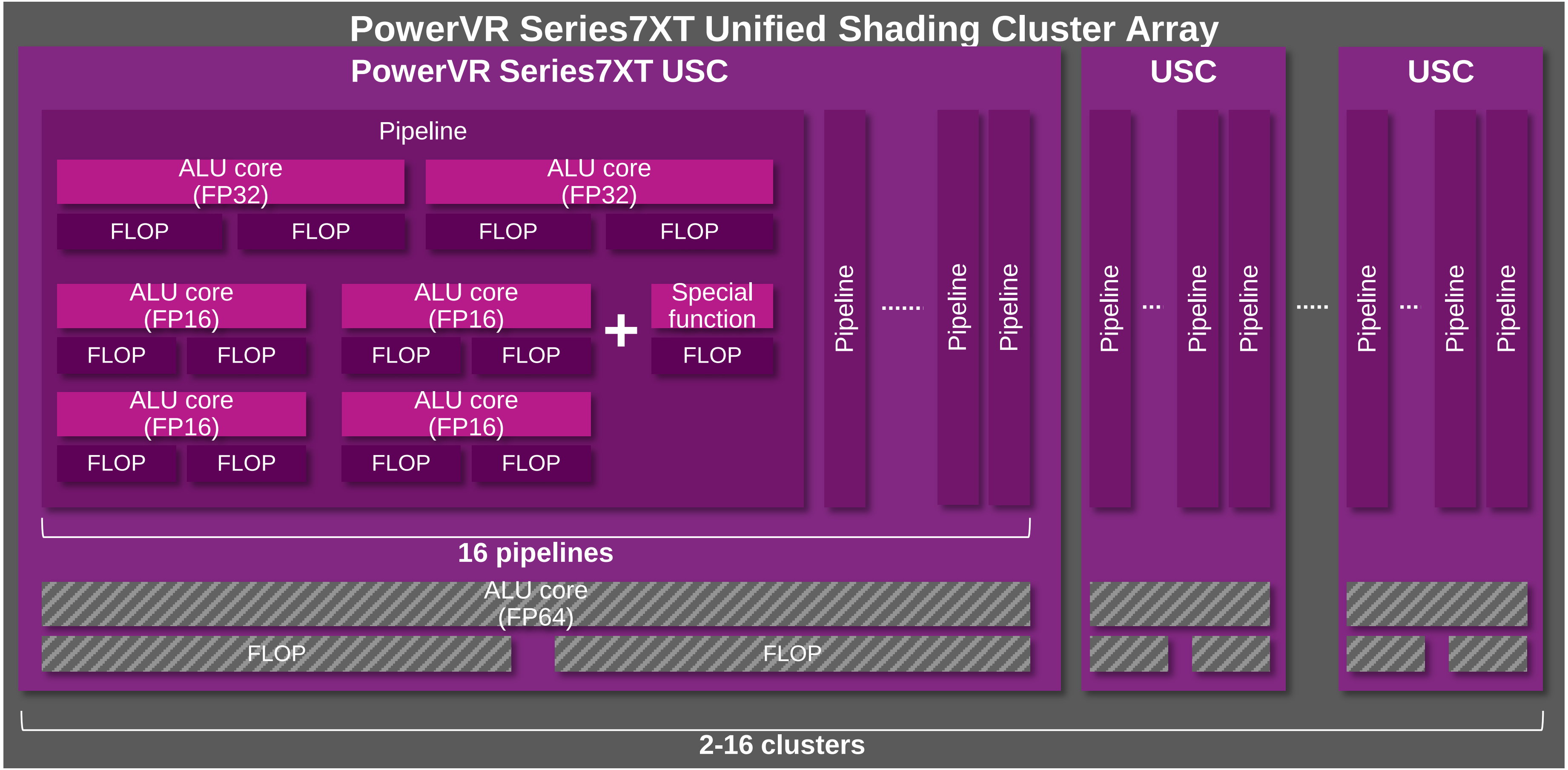
Lots of lessons learned

- Improved scheduler
- Streamlined ISA
- Improved compute task efficiency
- Completely new F16 datapath
- Improved front-end for sustained geometry performance
- ASTC

PowerVR Series7XT



▨ ***Extra low power GFLOPS***


PowerVR Series7XT Unified Shading Cluster Array
PowerVR Series7XT USC
Pipeline
ALU core (FP32)
FLOP
FLOP
ALU core (FP32)
FLOP
FLOP
ALU core (FP16)
FLOP
FLOP
ALU core (FP16)
FLOP
FLOP
+
Special function
FLOP
ALU core (FP16)
FLOP
FLOP
ALU core (FP16)
FLOP
FLOP
Pipeline
....
Pipeline
Pipeline
16 pipelines
ALU core (FP64)
FLOP
FLOP
USC
Pipeline
...
Pipeline
Pipeline
.....
USC
Pipeline
...
Pipeline
Pipeline
2-16 clusters

▨ *Optional*

Series6XT to Series7XT

Adding features and smoothing off rough edges

- **Changed how the architecture scales**
- **Improved USC**
- **Streamlined ISA**
- **Features**
  - Hardware tessellation
  - DX11-compliant USC (precision mainly)
  - FP64

Into the future

- **Exciting changes being worked on across the architecture**
  - USC
  - Front-end
  - Scaling
  - Stuff you want!

- **You can help**
  - We love feedback about the architecture and how it could best fit what you're doing
  - Don't be shy

Live Long, and Prosper

Imagination

www.imgtec.com/idc